

آکادمی رباتک

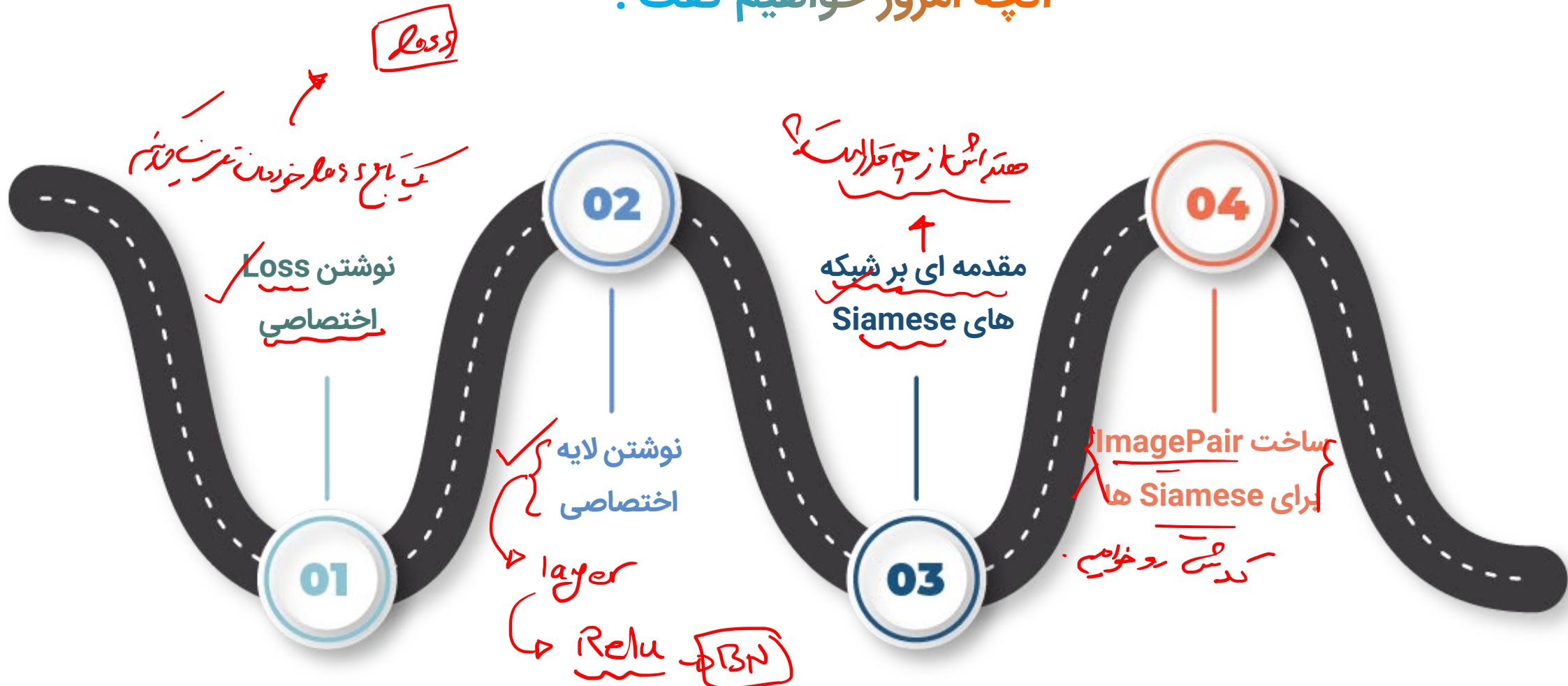
دوره تنسورفلو پیشرفته

جلسه سوم : Network Siamese و Loss Function



! پشتگرایی ✓
multiinput ✓
multi output ✓
:

آنچه امروز خواهیم گفت :



Keras ها در Loss Function استفاده از

```
net.compile(optimizer="adam", loss = "MSE")
```

میانگین مربعات خطای مطلق

Regression

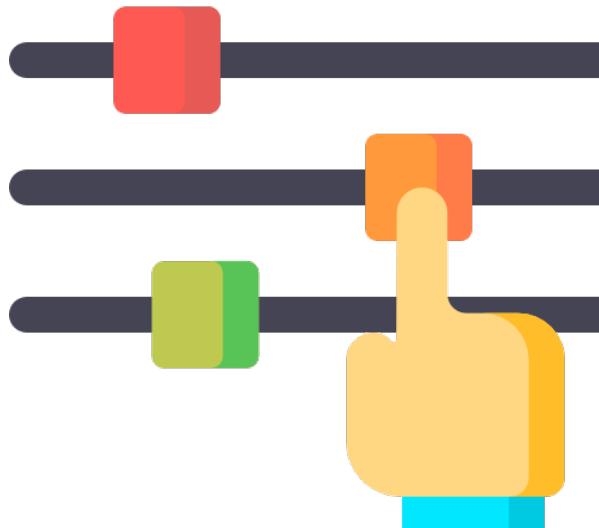
با

```
net.compile(optimizer="adam", loss=mean_squared_error)
```

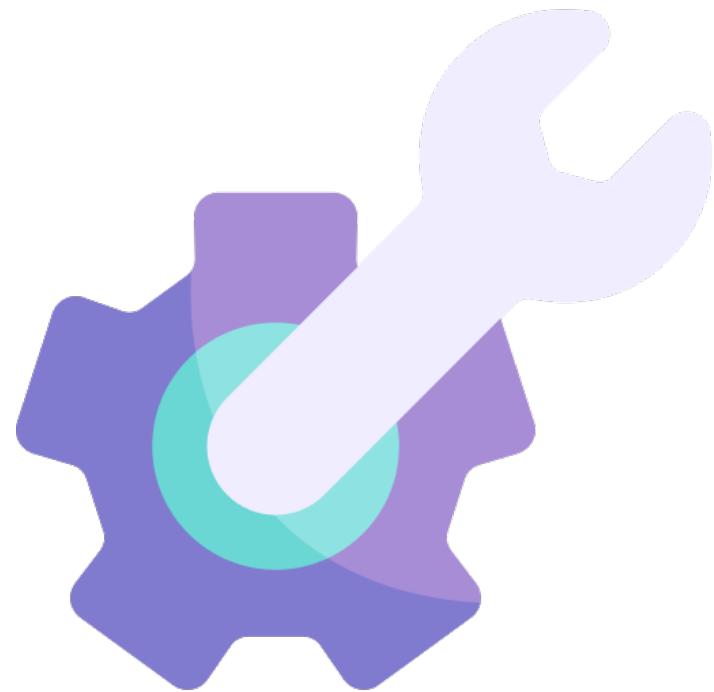
مزیتی حالتی که از Object استفاده می کنیم:

```
net.compile(optimizer="adam", loss=mean_squared_error(param = value))
```

هاست را این راسخ نمایم



اما اگر بخواهیم Loss اختصاصی خودمان را پیاده سازی کنیم چه؟ (بندگی)



برای اینجا

```
def my_loss_function(y_true, y_pred):  
    # حسنه برای خود  
    # ...  
    # ...  
    return losses
```

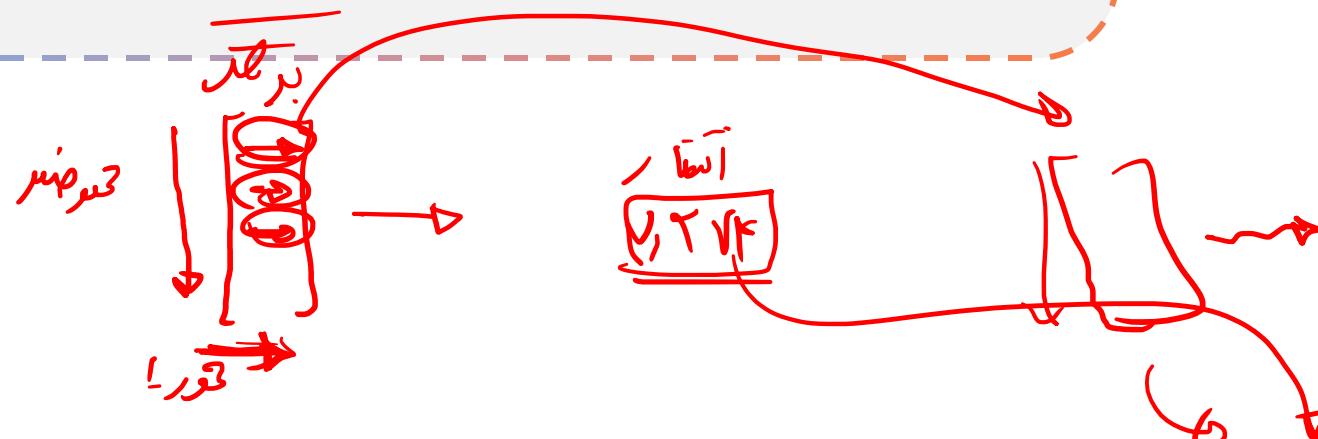
هر کدام خودی
حسنه

$y_{true} \Rightarrow$ مبلغ مطاع
 $y_{pred} \Rightarrow$ مبلغ پیش‌بینی

تابع MSE را پیاده سازی کنیم!

([0] → lst[-1]) آخرین هزار

```
def my_loss_function(y_true, y_pred):  
  
    error = tf.square(y_true - y_pred)  
    losses = tf.reduce_mean(error , axis = -1)  
  
    return losses
```



نتیجه
آخرین هزار
دستورات
شبکه عصبی

مثالی دیگر : Huber Loss

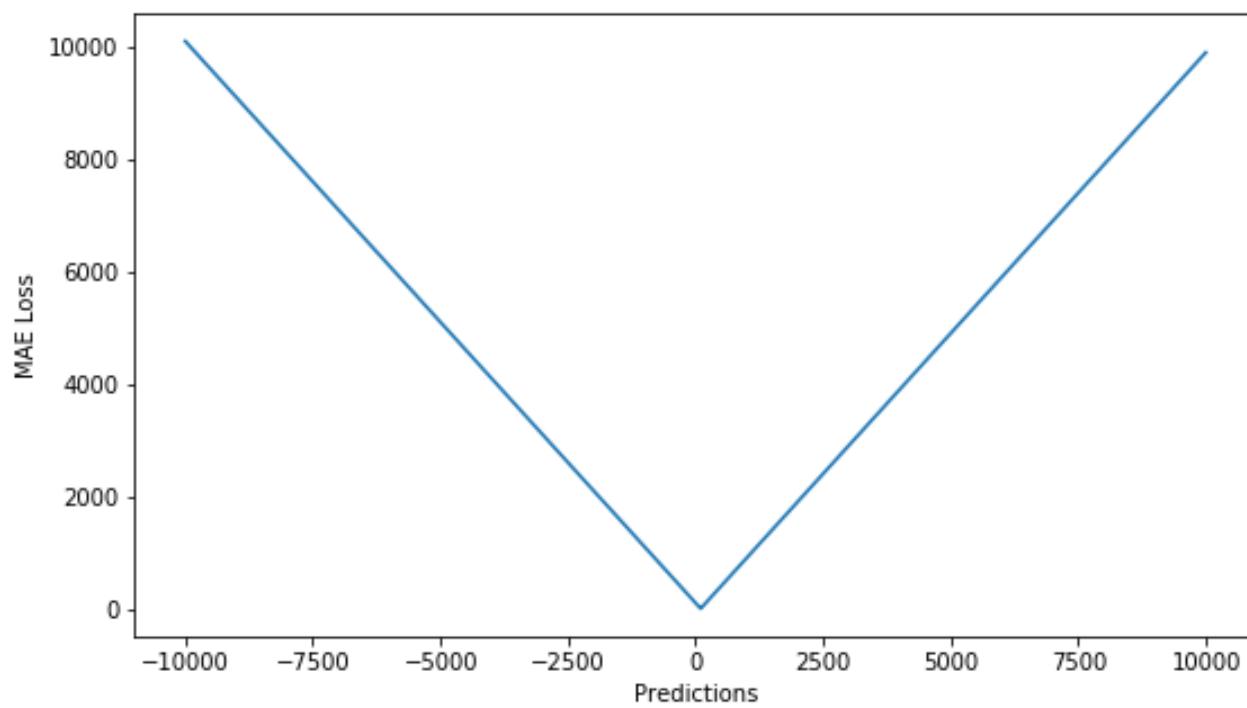
$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



ابتدا مروی بر

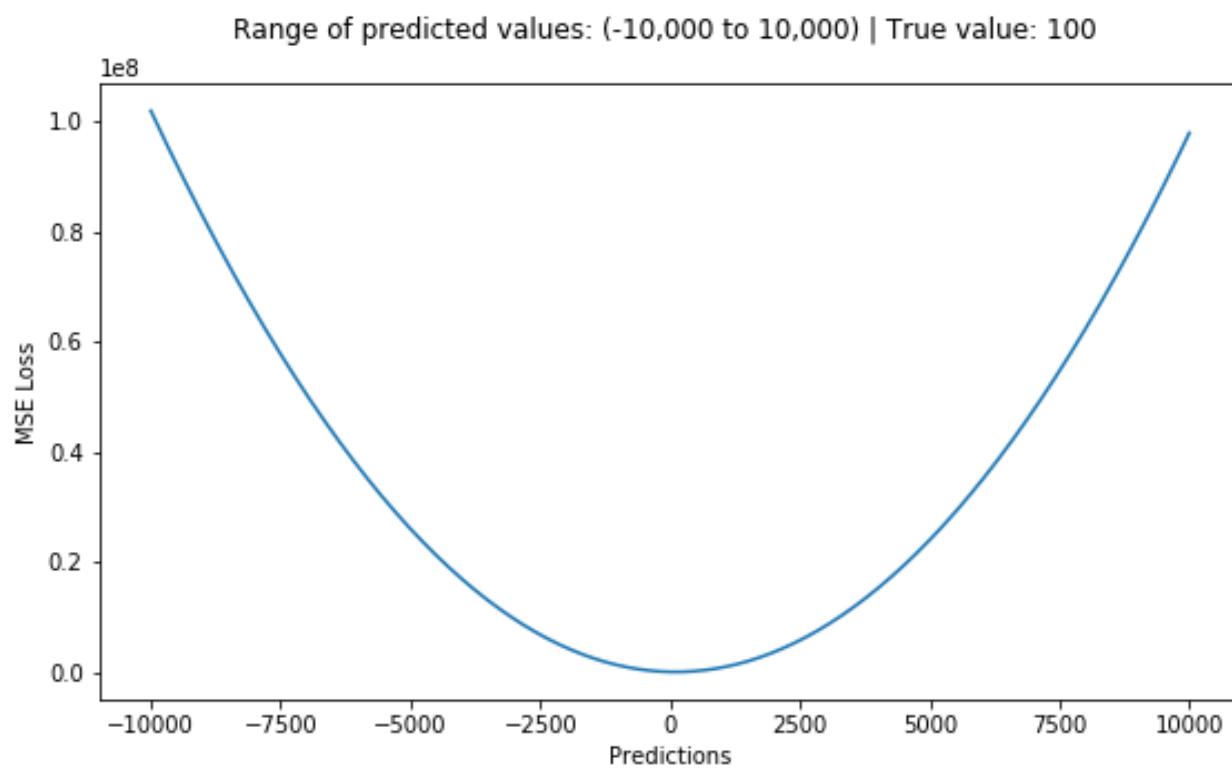
Mean Absolute Error

Range of predicted values: (-10,000 to 10,000) | True value: 100



$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

حال میوری بر Mean Squared Error



$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

مقایسه MSE و MAE در یک جمله

= =

بازی

حل مساله با MSE سریعتر است ولی با MAE پایدارتر. چرا؟

= =

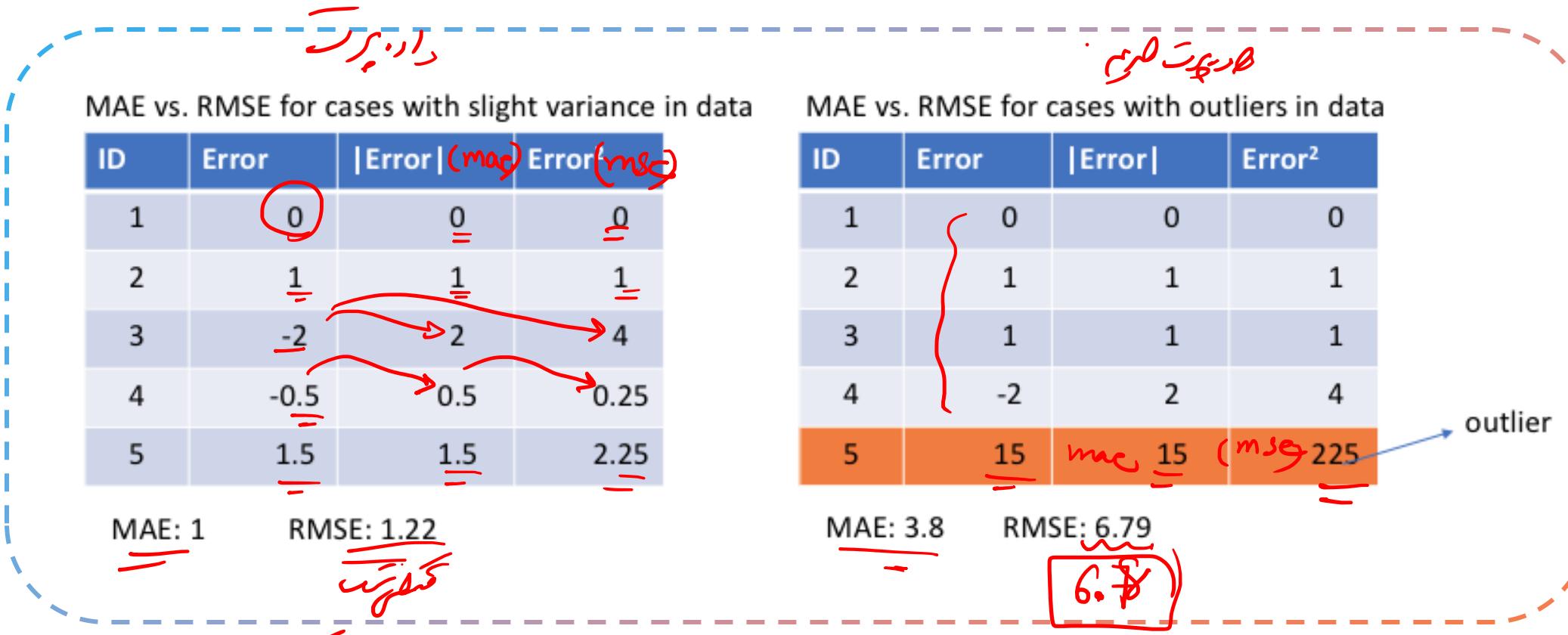
outlier

MSE → سریع‌تر است اما با خودکاری نیست

MAE → پایدار است



کمی دقیقتر همین موضوع را ببینیم:



حرف Huber Loss این است:

اگر میزان خطا از حدی کمتر بود از MSE استفاده کن و اگر بیشتر بود از MAE

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Small error
Big error
خطا کوچک
خطا بزرگ

حالا برویم سراغ پیاده سازی Huber Loss

```
def huber_loss(y_true, y_pred):  
    delta = 1  
    error = y_true - y_pred  
    is_small_error = tf.abs(error) <= delta  
  
    small_error_loss = tf.square(error) / 2  
    big_error_loss = delta * (tf.abs(error) - (0.5 * delta))  
  
    return tf.where(is_small_error, small_error_loss, big_error_loss)
```



کمی در مورد جزییات پیاده سازی حرف بزنیم:

خنبار را که رفته باش
با این خصوصیت های مخصوص برخواهیم

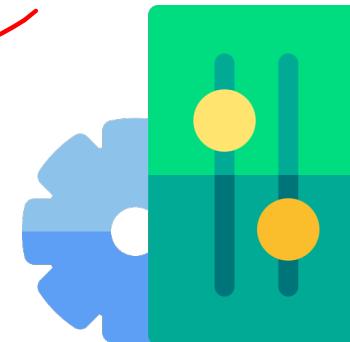
اگه بخواهیم پارامترها را پاس دهیم چه ؟



نحوه ارسال پارامتر به Loss

```
def huber_loss_with_delta(delta):  
    def huber_loss(y_true, y_pred):  
        error = y_true - y_pred  
        is_small_error = tf.abs(error) <= delta  
  
        small_error_loss = tf.square(error) / 2  
        big_error_loss = delta * (tf.abs(error) - (0.5 * delta))  
  
        return tf.where(is_small_error, small_error_loss, big_error_loss)  
  
    return huber_loss
```

برای معنی تعریف حالت های خود میرجست



استفاده از شبکه عصبی برای Loss

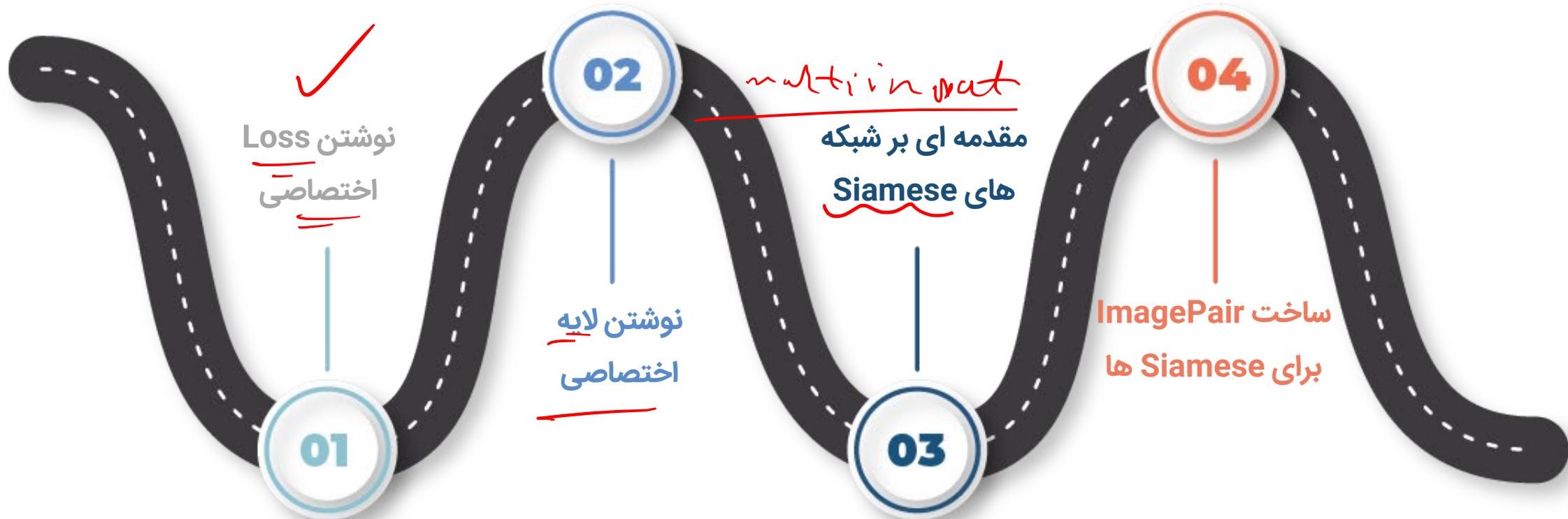
```
class MyHuberLoss(Loss):
    def __init__(self, delta):
        super().__init__()
        self.delta = delta

    def call(self, y_true, y_pred):
        error = y_true - y_pred
        is_small_error = tf.abs(error) <= self.delta

        small_error_loss = tf.square(error) / 2
        big_error_loss = self.delta * (tf.abs(error) - (0.5 * self.delta))

        return tf.where(is_small_error, small_error_loss, big_error_loss)
```

آنچه امروز خواهیم گفت :



نوشتن لایه اختصاصی در Keras



نوشتن لایه اختصاصی در Keras

```
y = layers.Conv2D(16, (3, 3), activation="relu")(y)  
y = layers.MaxPool2D((2, 2))(y)  
y = layers.Conv2D(64, (3, 3), activation="relu")(y)  
y = layers.MaxPool2D((2, 2))(y)
```

ReLU

```
def my_relu(x):  
    if x > 0:  
        return x  
    else:  
        return 0
```

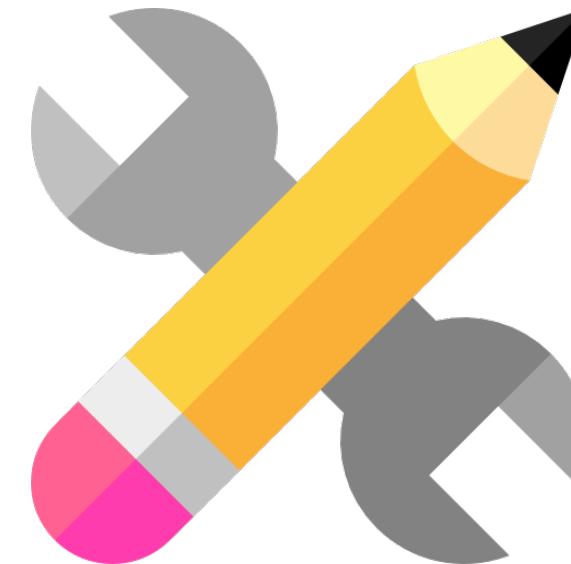
$\max(0, x)$

خودمان relu را پیاده سازی کنیم:

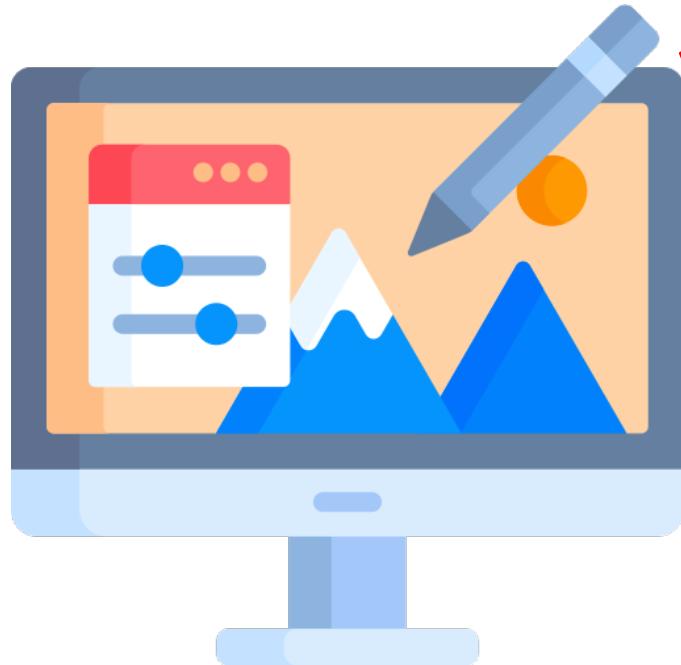
```
def my_relu(x):  
    return tf.maximum(0.0, x)
```

```
y = layers.Lambda(my_relu)(y)
```

برای این کار
برای این کار



حتی می توانیم آن را Customize کنیم!



$\text{max}(0, x)$

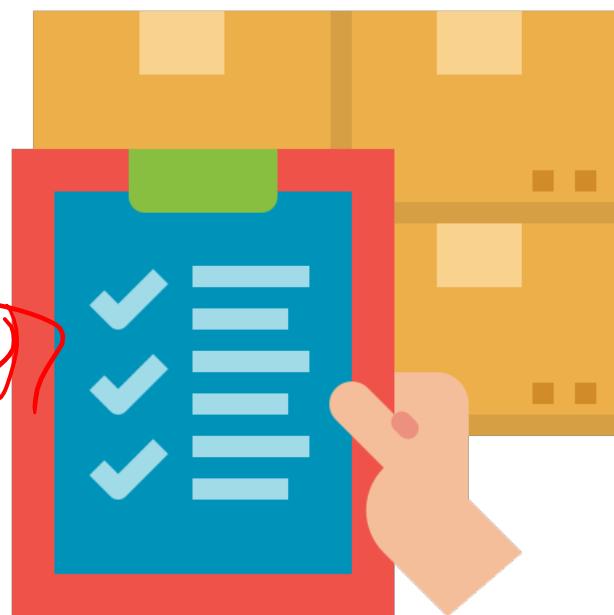
$0.5, x$

```
def my_relu(x):  
  
    return tf.maximum(0.5, x)  
  
y = layers.Lambda(my_relu)(y)
```

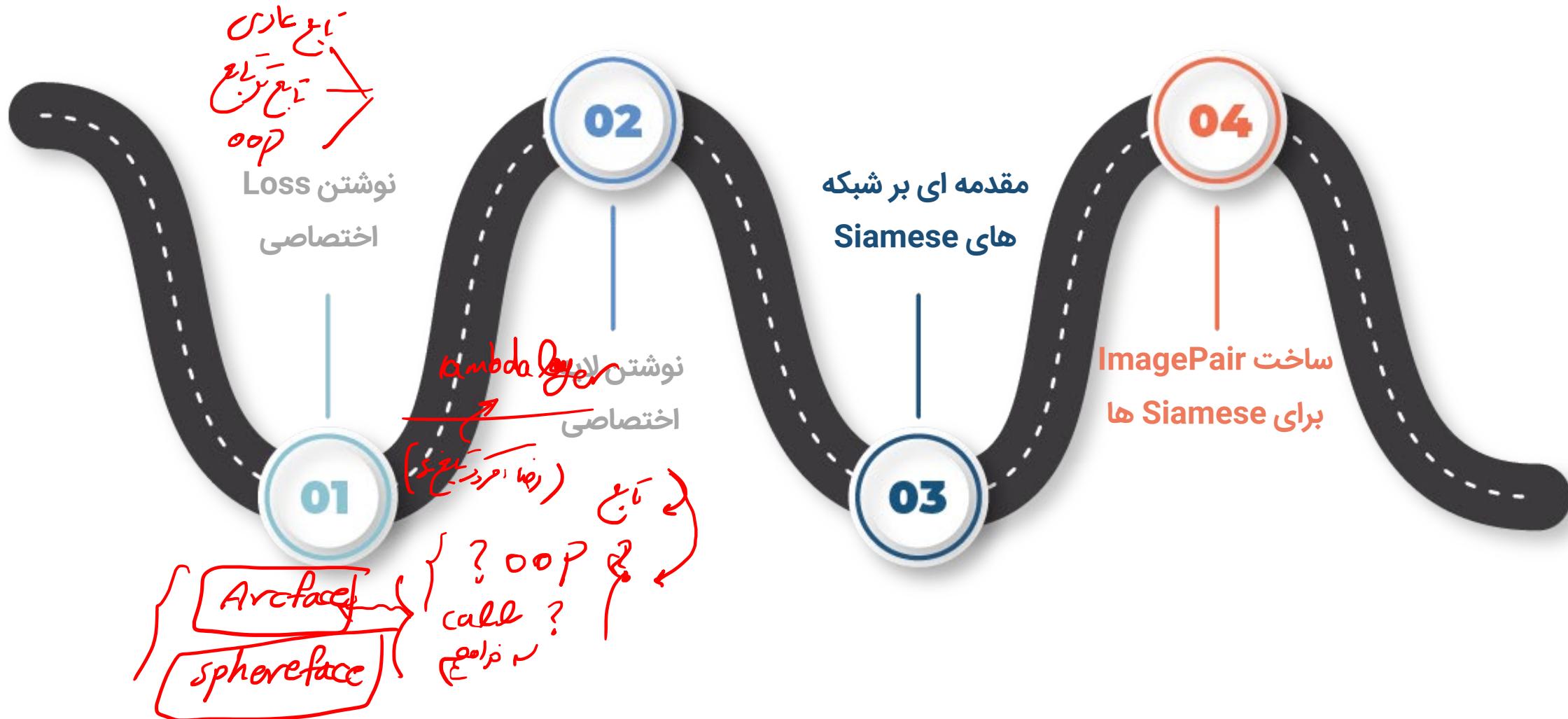
پیاده سازی کمی زیباتر

det test (t)
return y = layers.Lambda(lambda t: tf.maximum(0.0, t))(y)
def ~~t~~(~~y~~)

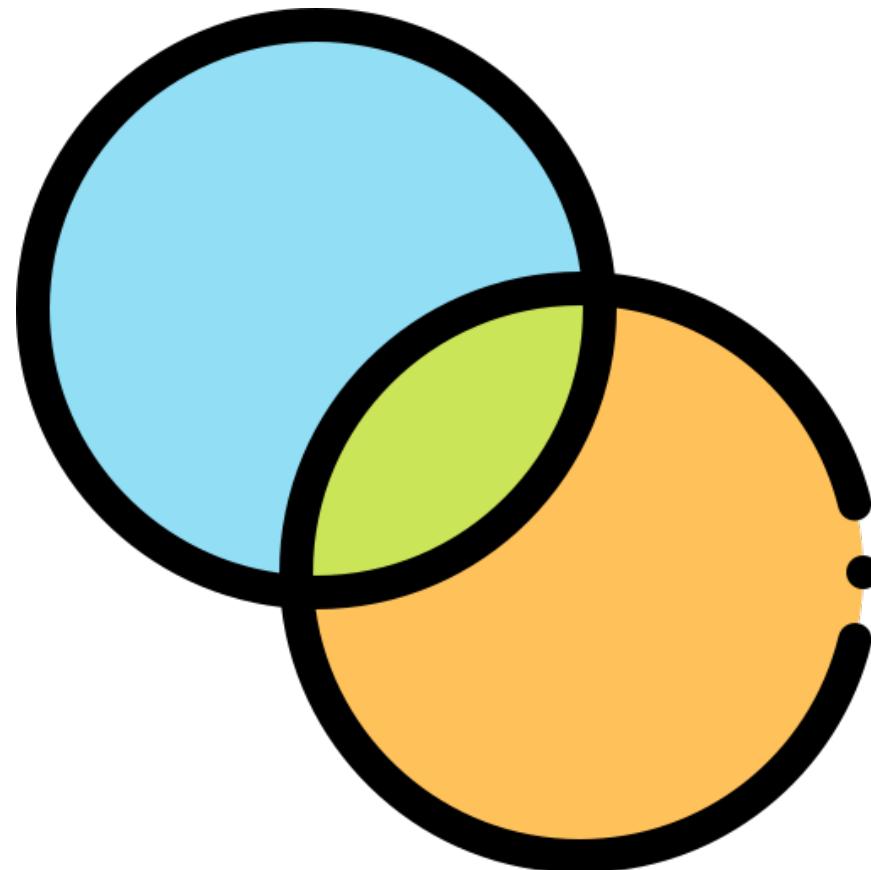
lambda خود چه ورودی
 t \rightarrow $tf.maximum(0.0, t)$



آنچه امروز خواهیم گفت :

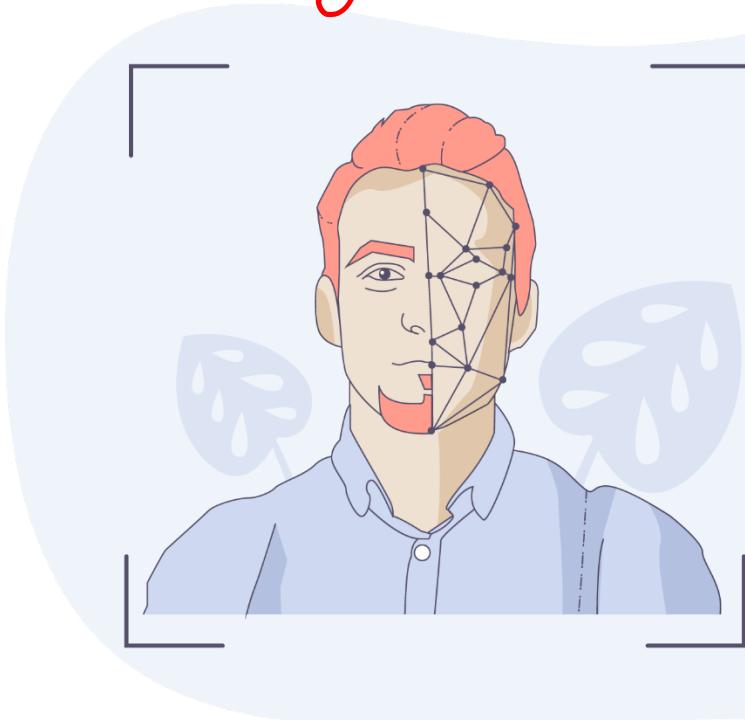


Siamese Network

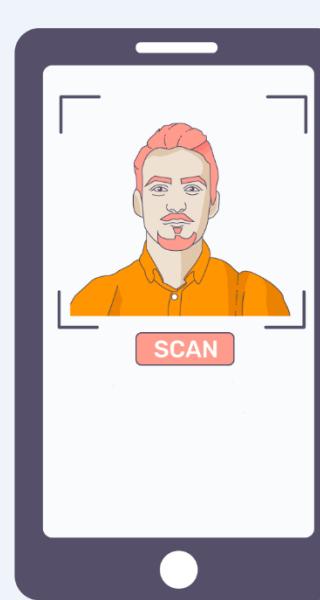


کاربردش کجاست ؟

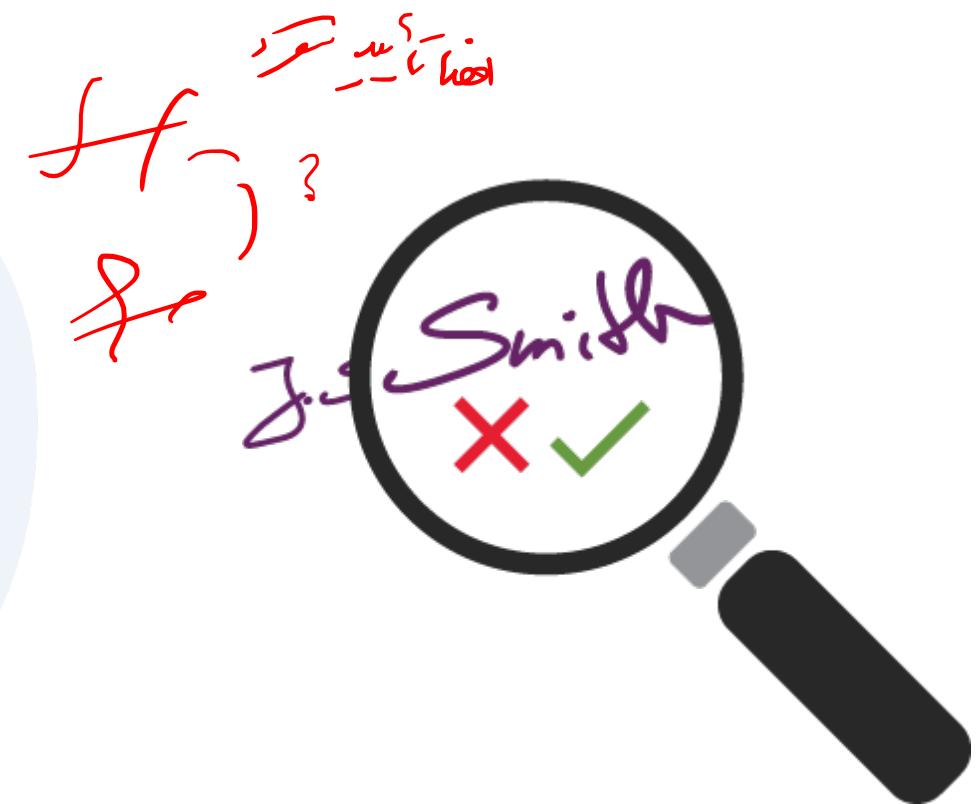
Verification
Recognition



Face Recognition

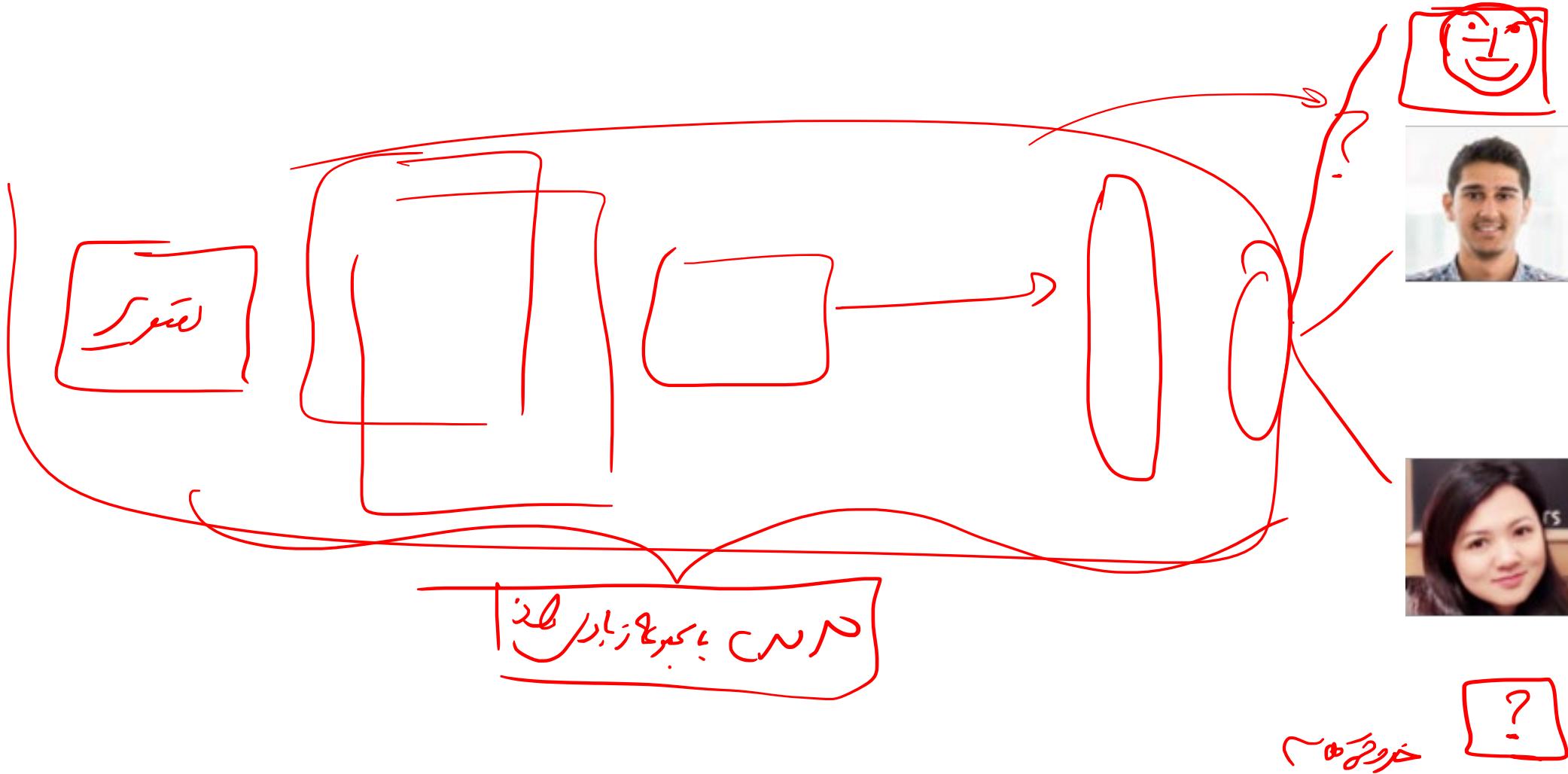


Correct

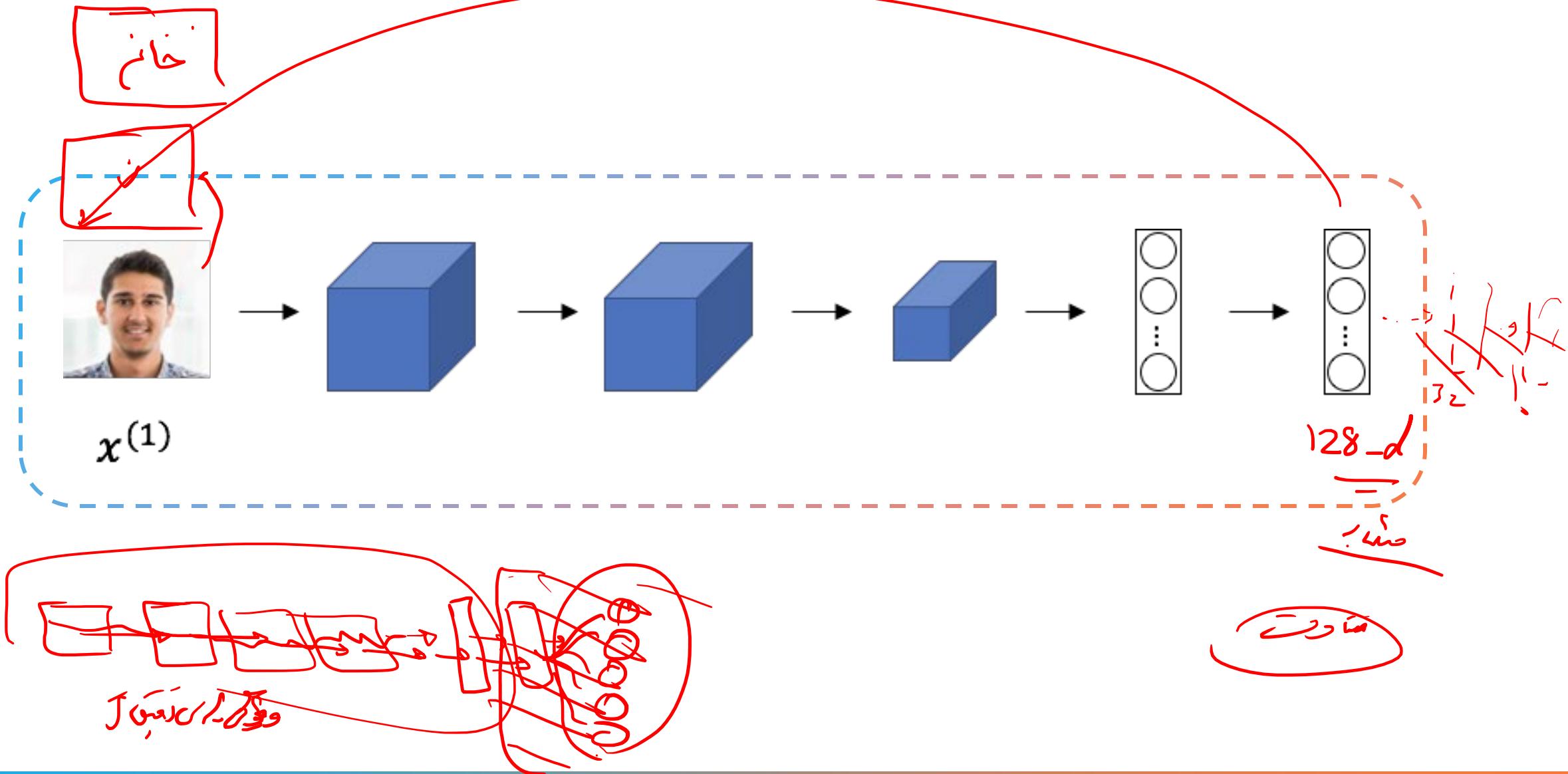


Signature Verification

باید مساله تشخیص چهره را به صورت کلاسیک حل کنیم:

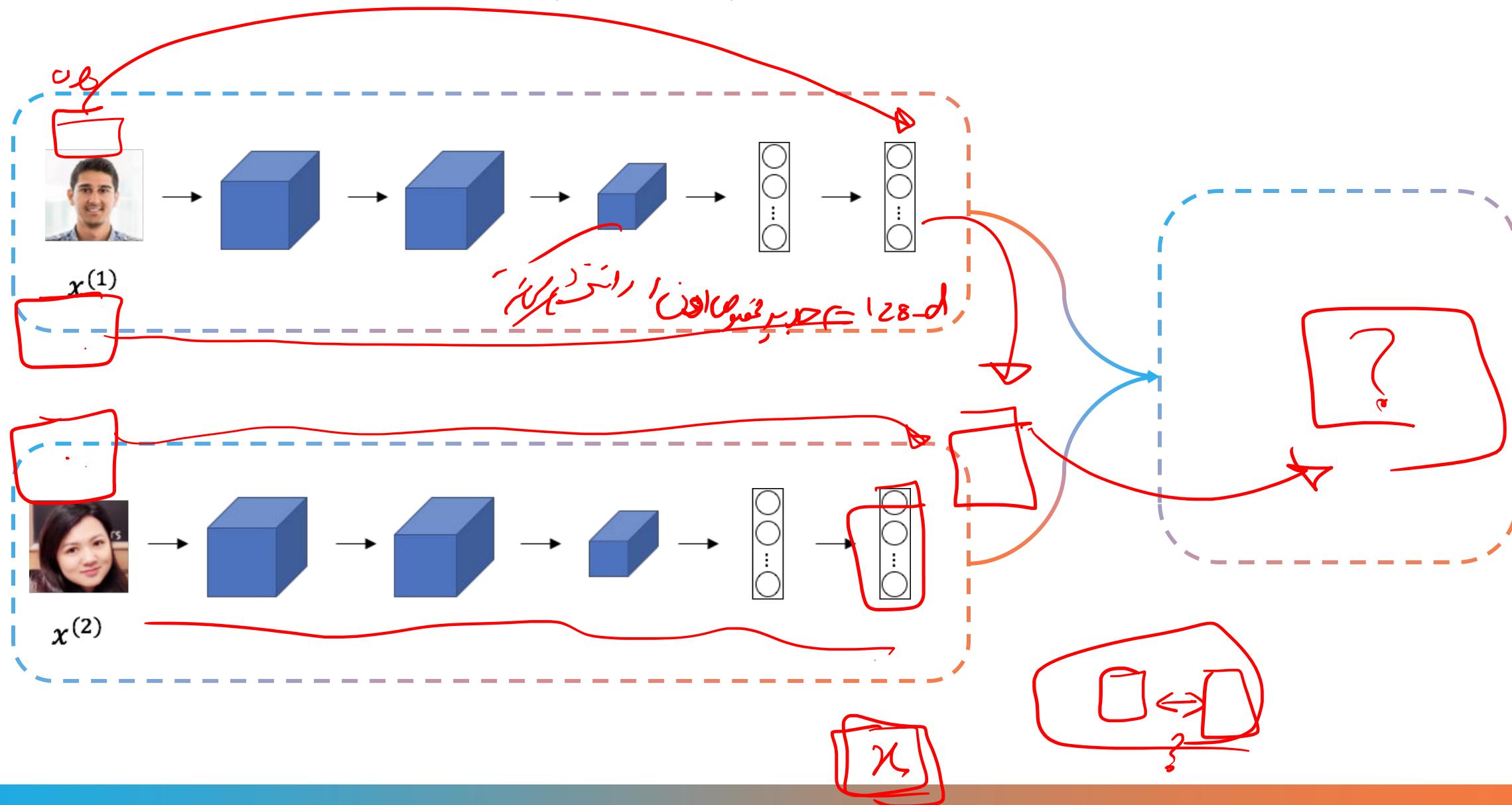


ابتدا مفهوم پردار Embedding



۱۵

حال می توانیم میزان شباهت دو چهره را این چنین محاسبه کنیم:



به این ساختار شبکه عصبی Siamese گفته می شود:

نوع خاص از شبکه (عصبی)

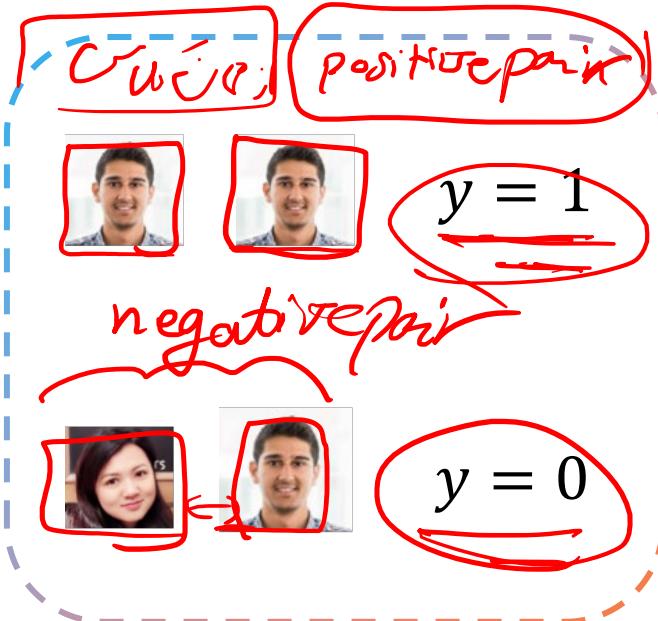


- این نوع شبکه از دو یا چند زیر شبکه تشکیل شده است.
- ساختار، پارامترها و وزن های این دو شبکه کاملاً کپی است.
- تغییر هر پارامتر در یکی از شبکه ها باعث تغییر در پارامتر دیگری می شود.

داده‌های

داده ما و لیبل ما چه اتفاقی برآش می‌فته؟

جنبه ورودی

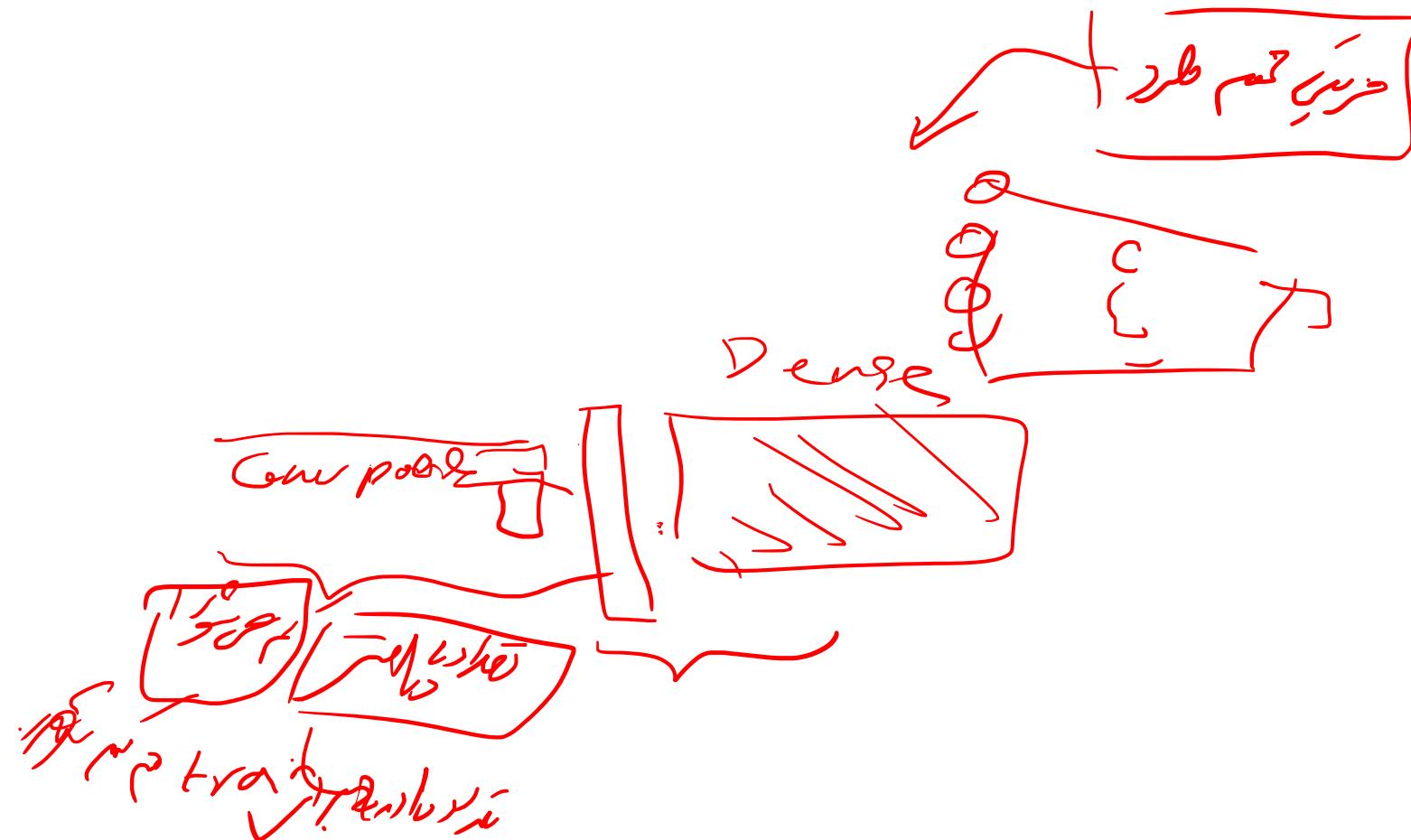
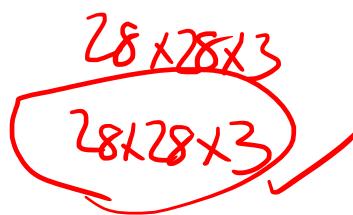
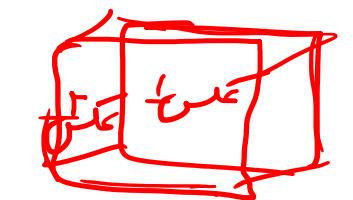


pair → / positive
negative

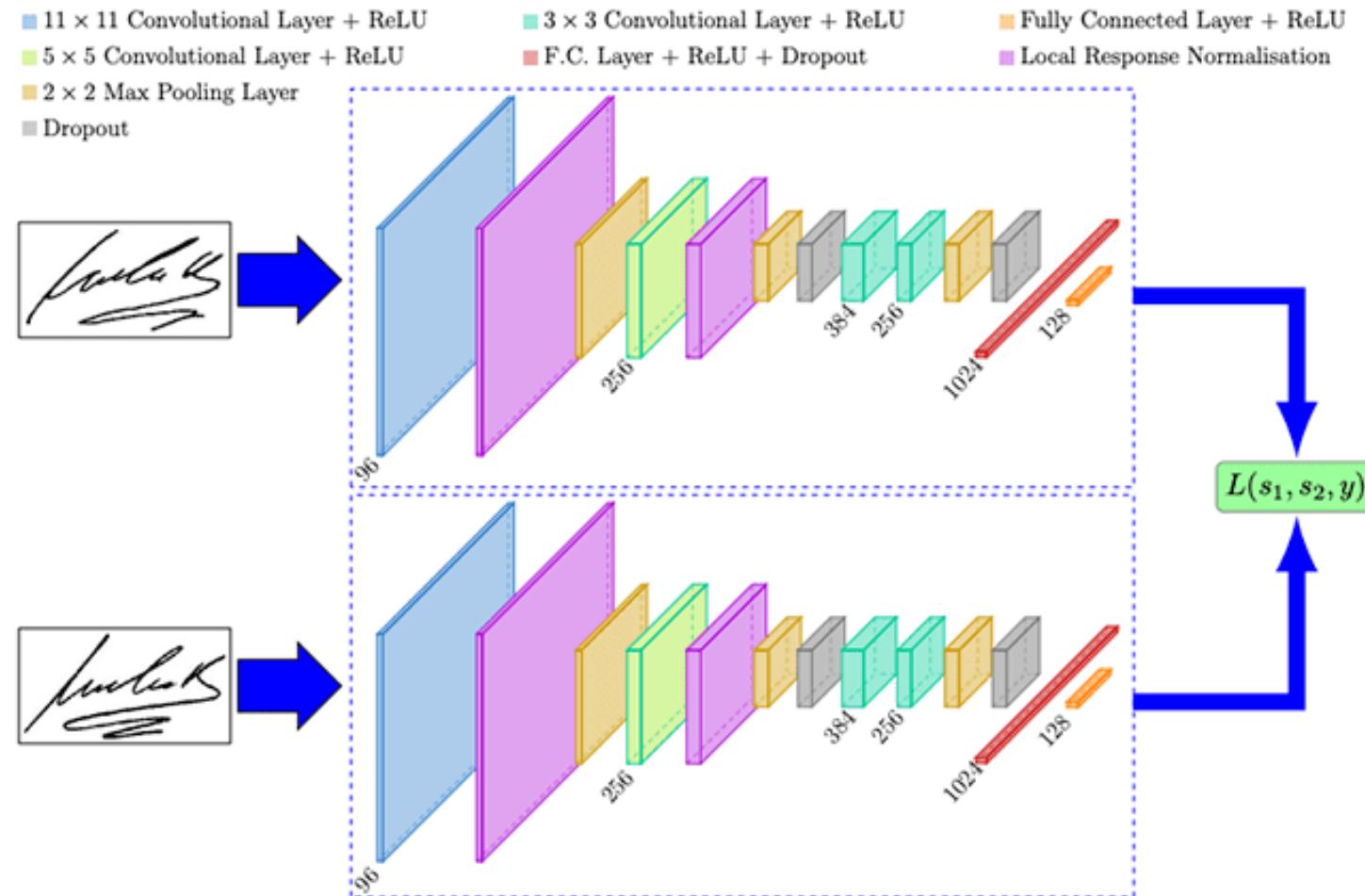
داده: یک ورودی این شبکه از دو داده مختلف تشکیل شده است.
= سه زیرساخت ورودی رئیس روانیس

لیبل ما یکی بودن یا نبودن داده های ورودی است.

این شبکه چه مزیتی نسبت به طبقه بندی عادی دارد؟



SigNet-2017



اما Train این شبکه چه ؟



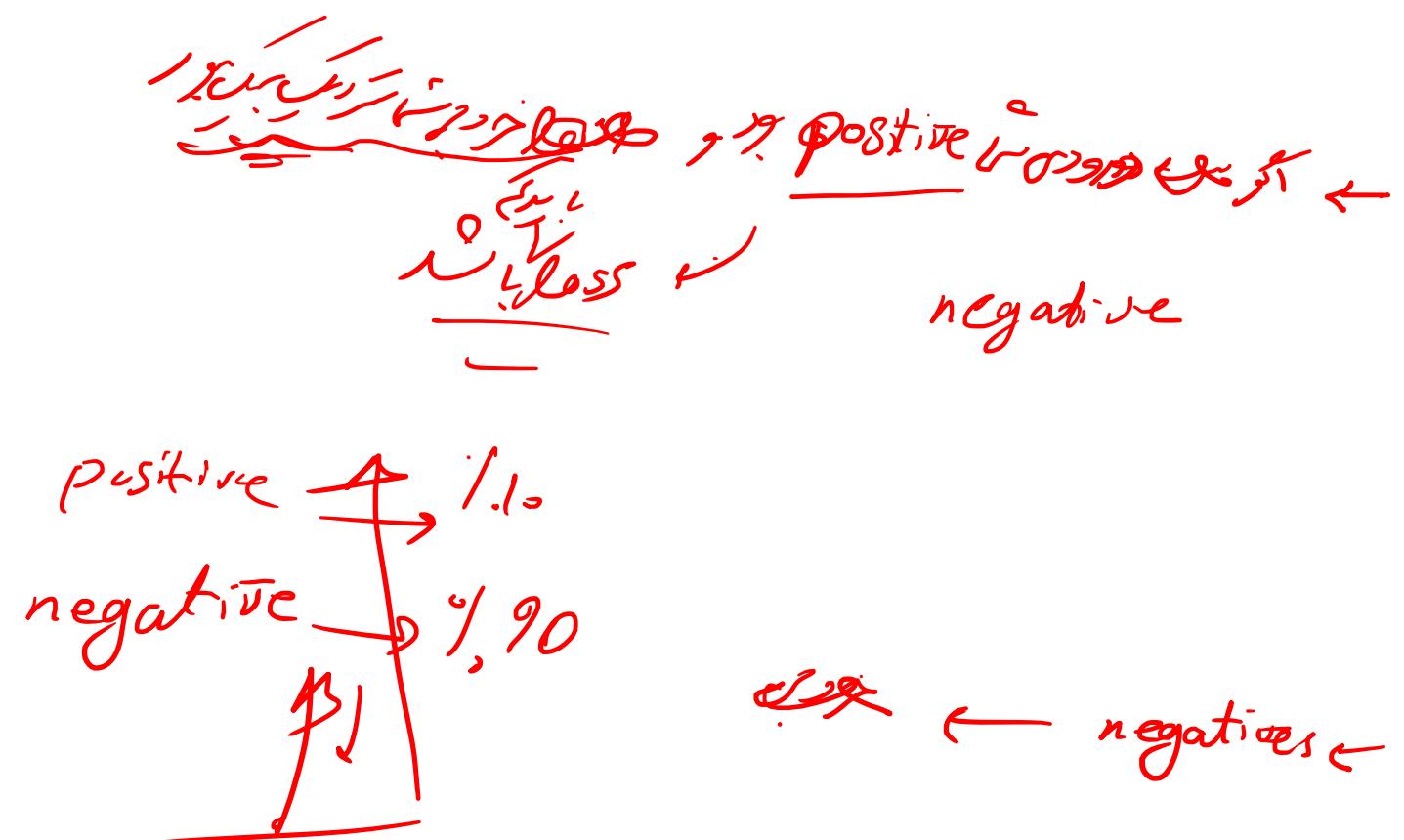
چه Loss Function ای برای آموزش این شبکه پیشنهاد می دهید ؟

لیبردرا



ابتدا یک یادآوری کوچک از ویژگی یک تابع Loss

contrastive loss



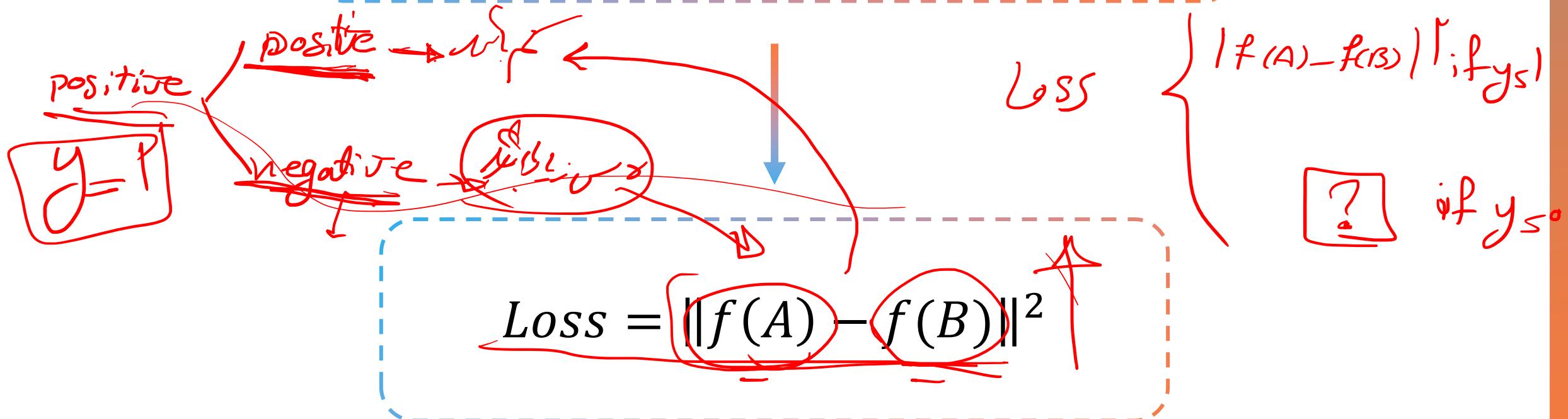
Contrastive Loss

Dimensionality Reduction by Learning an Invariant Mapping - 2006



Contrastive Loss

ابتدا حالتی که دو ورودی متعلق به یک کلاس باشند.



Contrastive Loss

حالی که دو ورودی از دو دسته مختلف هستند.



$$Loss = \circled{m} - \|f(A) - f(B)\|^2$$

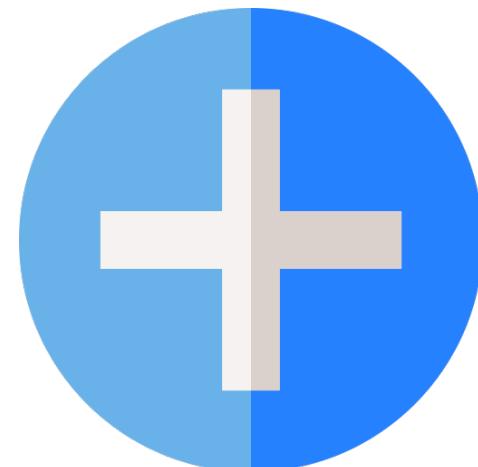
$$\circled{r} - \circled{r'} = \boxed{-1}$$

میانگین

طبیعی هست که Loss نباید عددی منفی باشد. بنابراین

$$\text{Loss} = \max(0, m - \|f(A) - f(B)\|^2)$$

$$; P_{y,1} \rightarrow P_{(A),1}$$



بنابراین خواهیم داشت :

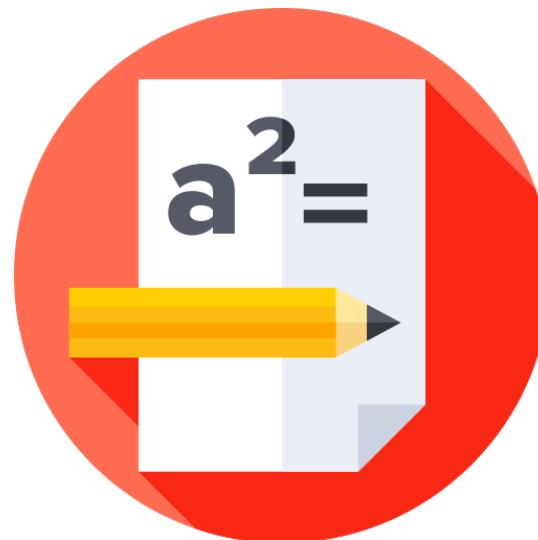
$$Loss = \|f(A) - f(B)\|^2$$

$$Loss = \max(0, m - \|f(A) - f(B)\|^2)$$

{ if $y = 1$

if $y = 0$

log loss



۳

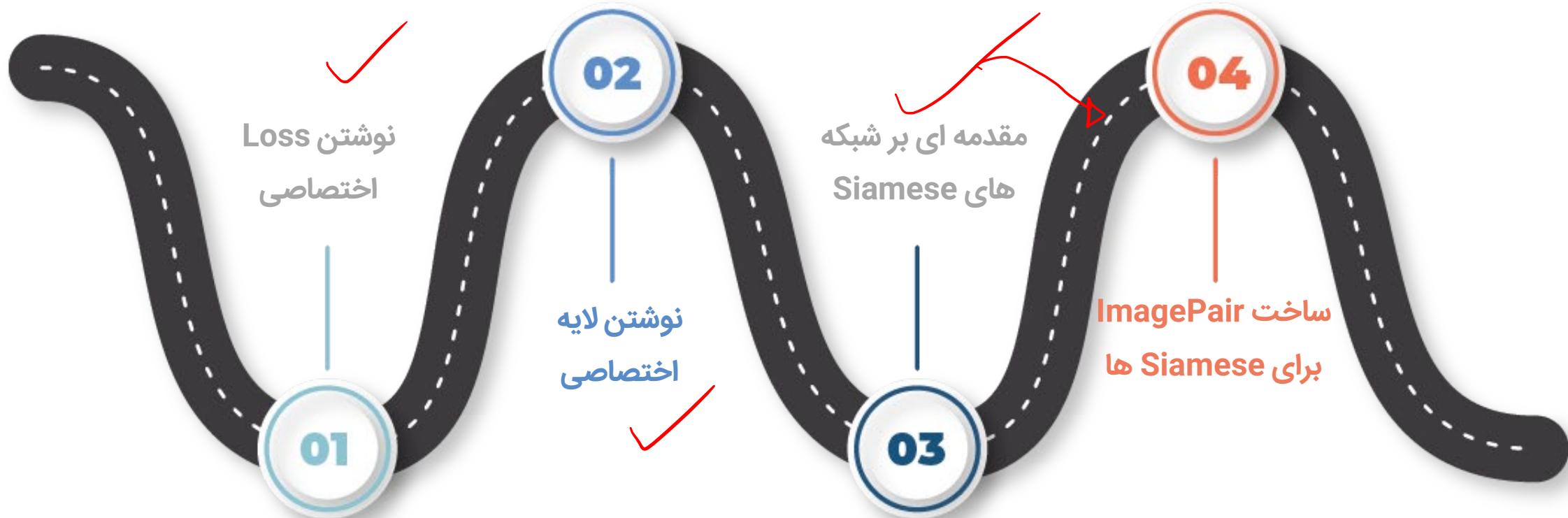
همان Loss کمی جمع و جور تر

لیکن
آنچه اینجا در اینجا میگویند

$$Loss = y \|f(A) - f(B)\|^2 + (1 - y) \max(0, m - \|f(A) - f(B)\|^2)$$



آنچه امروز خواهیم گفت :



حالا برویم سر وقت پیاده سازی **Siamese Network** ها

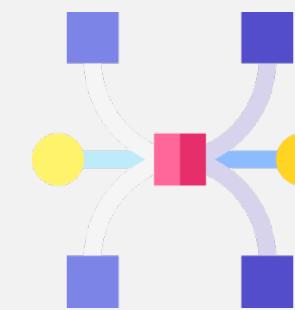


برای پیاده سازی شبکه Siamese باید چند کار انجام دهیم:



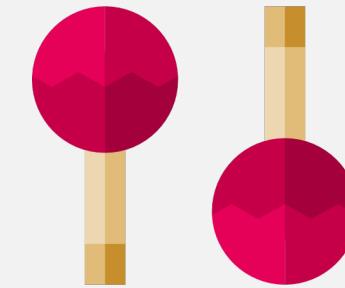
Loss Function

نوشتن
Contrastive loss



نوشتن لایه اقلیدسی

([] - []) که



ساخت Image Pair ها

positive pair
negative pair